REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
October 2011

2. REPORT TYPE
Technical Paper

3. DATES COVERED *(From - To)*
MAY 2010 – DEC 2010

4. TITLE AND SUBTITLE
LOCAL UTILITY ESTIMATION IN MODEL-FREE, MULTI-AGENT ENVIRONMENTS

5a. CONTRACT NUMBER
IN-HOUSE

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
61102F

6. AUTHOR(S)

Jeffrey Hudack (AFRL/RIEH)
Nathaniel Gemelli (AFRL/RISC)
Maria Scalzo (AFRL/RIEA)

5d. PROJECT NUMBER
230B

5e. TASK NUMBER
00

5f. WORK UNIT NUMBER
01

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/RIEA
525 Brooks Road
Rome NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIEA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-RI-RS-TP-2011-42

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA #: 88ABW-2010-6644
DATE CLEARED: 21 Dec 2010

13. SUPPLEMENTARY NOTES
This work is the result of a Department of the Air Force Office of Scientific Research Mini-Grant. The U.S. Government has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license to use, modify, reproduce, release, perform, display, or disclose the work by or on behalf of the Government.

14. ABSTRACT

Software agents are an enabling technology that supports rapid, automated, distributed decision making. Many joint task environments provide reward that is based on the performance of the collective, making it difficult to assign reward accurately to individual agents based on their performance. Some method is needed to assign the proper amount of credit to each of the agents in a collective, referred to as structural credit assignment, in an effort to maximize global utility. Within the multi-credit assignment problem the objective is to accurately estimate an agent's local utility based only on a global observation or global reward. To achieve an initial local estimate for each agent a Kalman filter technique is employed. The local utility estimates created through this technique however are independent of knowledge held by other agents in the environment. This leads to the intuition that there is room to improve local utility estimation through the sharing of knowledge between agents. Hence, different communication schemes are explored in order to not only improve the local estimates provided by the Kalman filter but in an effort to allow the agents to more rapidly converge to good policies.

15. SUBJECT TERMS
Kalman FIlter, Multi-Credit Assignment, Local Utility Estimation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					MARIA SCALZO
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER (*Include area code)***
U	U	U	UU	13	N/A

Local utility estimation in model-free, multi-agent environments

Jeffrey Hudack
Air Force Research Laboratory / RIEH

Nathaniel Gemelli
Air Force Research Laboratory / RISC

Maria Scalzo
Air Force Research Laboratory / RIEA

December 1, 2010

Abstract

Software agents are an enabling technology that supports rapid, automated, distributed decision making. Many joint task environments provide reward that is based on the performance of the collective, making it difficult to assign reward accurately to individual agents based on their performance. Some method is needed to assign the proper amount of credit to each of the agents in a collective, referred to as structural credit assignment, in an effort to maximize global utility. Within the multi-credit assignment problem the objective is to accurately estimate an agent's local utility based only on a global observation or global reward. To achieve an initial local estimate for each agent a Kalman filter technique is employed. The local utility estimates created through this technique however are independent of knowledge held by other agents in the environment. This leads to the intuition that there is room to improve local utility estimation through the sharing of knowledge between agents. Hence, different communication schemes are explored in order to not only improve the local estimates provided by the Kalman filter but in an effort to allow the agents to more rapidly converge to good policies.

Contents

1 Introduction 4
1.1 Multi-agent reinforcement learning 4
1.2 Collective Intelligence 4
1.3 Prior Work 5

2 Methods, Assumptions, and Procedures 5
2.1 Kalman Filter for reward estimation 5
2.2 Communication Schemes 6
2.2.1 Type I 7
2.2.2 Type II 7
2.2.3 Type III 7
2.2.4 Type IV 7
2.2.5 Type V 7
2.3 Test Environments 8

3 Experimental Results 8

4 State Value Estimation 10

5 Conclusion 10

6 Symbols, Abbreviations and Acronyms 12

List of Figures

1 5x5 Hop World environment 8
2 5 x 5 Cliff World environment 8
3 Performance of communication schemes on HopWorld (30 runs) 9
4 Performance of communication schemes on CliffWorld (30 runs) 9

1 Introduction

As computational hardware becomes smaller and more ubiquitous there is an increasing need for algorithms that can be effectively distributed across a collective of independent platforms. These individual elements, often referred to as agents in the machine learning community, should exhibit both local autonomy as well as contribute to a global goal structure that benefits the system as a whole.

Joint task environments often provide a reward that is based on the performance of the collective, making it difficult to assign reward accurately to individual agents based on their performance. While it's possible to design an environment with component rewards the task becomes difficult when there are a large number of agents or complicated tasks for which the pareto-optimal solution is not clear. If we hope to apply distributed learning methods to a wider range of problems it is necessary to find methods for determining reward distributions for multiple independent learners.

We focus on cooperative, model-free multi-agent environments that provide a team-based global utility based on the performance of all agents. We use the term model-free in the sense that the only information an individual agent has to make a decision is its local state and the global reward(s) provided by the environment. While additional knowledge could improve performance we first seek to find a domain independent solution that does not rely on this a priori knowledge.

1.1 Multi-agent reinforcement learning

Reinforcement Learning (RL) [5] is a sub-area of machine learning that concerns itself with learning what action to take in a given state of an environment. In doing so, the goal is to maximize a utility provided by the environment. Reinforcement learners are not given direct instruction on how to accomplish a given task, instead learning a policy through iterative experience. By interacting with the environment the learner will build a mapping of actions to states that give it the highest expected return in the form of a long-term reward. Generally, we represent the learning problem in terms of a Markov Decision Process (MDP), so obtaining high (or even optimal) expected return becomes a problem of solving the MDP that represents the environment an agent is operating in.

When a reward signal is provided by an environment we must ask, how do we distribute credit to each action in the current episode which lead to the reward signal it received? This challenge, known as the credit assignment problem, lies in the proper distribution of reward to the actions that contributed to the solution. In single agent learning temporal credit assignment is used to recognize how much individual actions, as part of a series of actions, contribute to local and global rewards. This interest has led to the development of a family of reinforcement learning algorithms called temporal differencing methods [5] that have proven to be very effective.

Multi-agent systems introduce a new set of challenges to the unsupervised learning process that are not present in single agent environments. Since rewards are a product of the actions of multiple agents it can often be difficult to determine which actions performed by each agent contributed to the reward. Therefore, some method is needed to assign the proper amount of credit to each of the agents in a collective in an effort to maximize global utility. This is especially difficult because many environments that involve multiple agents require the maximization of a single global reward. An attempt to unify temporal and structural assignment has been proposed in [1] but relies on the assumption that the actions taken by individual agents are not concurrent and can be ordered serially.

1.2 Collective Intelligence

The COIN (COllective INtelligence) framework, summarized in [6], describes a set of relations between world utility and local utility that we use to characterize agent interactions with both the environment and each other. This relationship is defined by two properties referred to as factoredness and learnability, both of which provide the basis for our description of learning environments.

Factoredness represents the influence that an agent has on the reward, meaning that when the local reward increases as should the global reward. Formally, if all other local rewards are fixed an increase in local reward for one agent should never result in a decrease in the global reward. This property insures that individual agents can use the global utility as an indication of their local performance even if it's obscured by the local rewards of the other agents in the system.

Learnability indicates to what degree the global reward is sensitive to an individual agent's choices as opposed to determined by the other agents in the system. This represents the ratio of signal to noise that each agent must deal with. As the number of agents in the system increases the learnability decreases. In severe cases the individual contribution to the reward could be so small that it's indiscernible from minor noise. In [4] two agents use RL to coordinate the pushing of a block without any knowledge of each other's existence. While this is similar in the sense of being model-free it only involves two agents, which provides for high learnability.

1.3 Prior Work

Our previous work [7] showed that the Kalman filter proved to be an effective means of state value estimation in simple environments but degraded in performance as the number of agents increased and, conversely, the learnability decreased. Due to the fact that as more agents are added to the environment, more noise is added to the global reward and hence it becomes more difficult for each individual agent to discern its own contribution to the performance of the system as a whole. As a result, our focus shifted to finding methods of communicating limited information to increase the information each agent has with minimal use of bandwidth.

Additionally, exploration of the various Kalman filter parameters led to the conclusion that the performance, both individually and as a collective, was not sensitive to these parameters except at extreme values. This leads us to believe that the Kalman filter may be in excess of what is necessary to make these estimations and another, simpler solution may exist that is just as effective. With this in mind we have explored a fully centralized linear estimator with methods for estimating hidden states. By approaching this problem using both fully distributed and centralized methods we hope to find an effective middle ground solution that will leverage the strengths of both while minimizing the amount of communication required.

2 Methods, Assumptions, and Procedures

2.1 Kalman Filter for reward estimation

The Kalman Filter (KF) [3] is an optimal (unbiased) estimator for problems with linear Gaussian characteristics. A recursive fiter, the KF estimates the true state of a system based on numerous observations of the state. The algorithm is quite simple, producing estimates by performing two stages, prediction and update. In the prediction phase, the state and covariance estimates from the previous iteration are projected forward to the current observation period using state transition and process noise covariance matrices. Next, the update stage, corrects the predicted state and covariance estimates through a weighting factor known as the Kalman gain. This process is repeated iteratively over the entire observable period of the state.

A Kalman Filter was used in [2] to generate local utility estimates based on the global utility received at each time step. In essence, this approach creates a mapping from states to rewards based on the variance of the global utility with respect to the states visited. As a result each agent forms a more accurate estimation of local utility for each state through repeated visits and varying global utilities. This approach does not only function without any models of the agents or environment it doesn't even need to know that the other agents in the system even exist. This makes it a powerful method of estimation usable in a wide range of tasks.

In this approach, the Kalman Filter is used to generate estimations of state utilities from noisy data. The global reward at time t is a linear combination of the local reward for being in state i and a noise term b at time t :

$$g_t = r(i) + b_t \tag{1}$$

Here the term $r(i)$ represents the factoredness property, or the influence that an agent has on the global reward for being in state i. b_t is the contributions made to the global reward at time t by all other agents.

This can also be interpreted as a signal $r(i)$ with some noise b_t. Given a set of world states $1...N$ we can model agent a's state estimations as follows:

$$x_t^a = \begin{bmatrix} r_t^a(1) \\ \vdots \\ r_t^a(N) \\ b_t^a \end{bmatrix} \tag{2}$$

where $r_t^a(i)$ is agent a's estimate of the reward for being in state i at time T. The observation and state transition equations are respectively:

$$g_t = C^a x_t^a \tag{3}$$

$$x_t^a = x_{t-1}^a \tag{4}$$

The state measurement transformation matrix of agent A is:

$$C^a = [0 \dots 1_i \dots 01]\,, \tag{5}$$

where 1_i is the i^{th} index of the matrix, with i being the current state of agent a.

2.2 Communication Schemes

Based on the definition in the previous section it is clear that each agent creates its own local estimate of the world. In cases where each agent may have a different value for a given state this type of estimation is necessary. However, in instances where the reward for being in each state is common among all agents, one would expect communication between agents and subsequent combination of the agents' local estimates to converge more quickly to a correct reward estimation.

Communication would allow an agent that has not yet visited state i to benefit from another agent's knowledge of the value of i through an exchange of state information. However, it is not beneficial to simply exchange this information because an agent may communicate an incorrect estimate and the exchange does not take into account the individual history that led each agent to its estimation. It is important that state information be exchanged in a way that can improve each agent's beliefs about the value of states in the world.

We explore the utility of exchanging select pieces of state information, as well as different techniques for combining this information. The schemes described in this section all involve communication between randomly selected pairs of agents chosen at each time step.

When the Kalman filter is initialized it is often biased by the first global reward it receives. This leads to a set of estimates that may not be accurate representations of the state environment, but the relative value of the estimates are still intact. For example, consider an environment that has all 0 reward states with one state giving a reward of 20. One agent may estimate that all states are worth 20 with one state offering a reward of 40, while another agent may estimate all states are worth -10 with one state being worth 10. Because the reinforcement learner bases all decisions on the differential between state/action pairs, both representations will work equally as well. However, when the agents are sharing information it's important to normalize this difference as part of the information exchange process.

To address this discrepancy the agents compute the offset between the mean of all of their estimates and use this value to make their estimates comparable in a meaningful context. To represent this offset from the perspective of agents a and b we define:

$$\Omega_a = \frac{1}{N}\sum_{i=0}^{N} r_t^a(i) - \frac{1}{N}\sum_{j=0}^{N} r_t^b(j) \tag{6}$$

$$\Omega_b = \frac{1}{N}\sum_{j=0}^{N} r_t^b(j) - \frac{1}{N}\sum_{i=0}^{N} r_t^a(i) \tag{7}$$

2.2.1 Type I

Agent i and j exchange their current state at time t. Each agent runs an additional iteration of the Kalman filter at each time step, updating their own state estimates twice per time step. This does not require use of the offsets because each agent is using the global reward to update its estimates.

2.2.2 Type II

At each time step a random selection of agent pairings exchanges their current state as well as their current estimate of that state's value. This information is used to compute a weighted sum based on the "experience" that each agent has for that state. This is used to update the estimate of the corresponding state for each agent.

Let $cnt_t^a(k)$ be the number of times agent a has visited state k prior to time t. Let agent a be in state i and agent b be in state j at time t. Agent a shares $r_t^a(i)$ and $cnt_t^a(k)$ and agent b responds with $r_t^b(k)$ and $cnt_t^a(k)$. Agent a will update its estimate of $r_t^a(k)$ according to:

$$\hat{r}_t^a(j) = (r_t^a(j) - \Omega_a)\left(\frac{cnt_t^a(j)}{cnt_t^a(j) + cnt_t^b(j)}\right) + r_t^b(j)\left(\frac{cnt_t^b(j)}{cnt_t^a(j) + cnt_t^b(j)}\right) + \Omega_a \tag{8}$$

Similarly, agent b will updates it estimate of $r_t^b(k)$ with the information received from agent a as follows:

$$\hat{r}_t^b(i) = (r_t^b(i) - \Omega_b)\left(\frac{cnt_t^b(i)}{cnt_t^a(i) + cnt_t^b(i)}\right) + r_t^a(i)\left(\frac{cnt_t^a(i)}{cnt_t^a(i) + cnt_t^b(i)}\right) + \Omega_b \tag{9}$$

2.2.3 Type III

Much like Type II the agents exchange information regarding their current state. As an additional step the agents then share their estimates for both the state they are in and the state that their partner is in. This provides updates for two states with each exchange. It is noted that this would require two exchanges, the first being the sharing of the current states i and j of agent a and b respectively, the seconds being the pair of estimates for states i and j from each agent. The two new updates are as follows:

$$\hat{r}_t^a(i) = (r_t^a(i) - \Omega_a)\left(\frac{cnt_t^a(i)}{cnt_t^a(i) + cnt_t^b(i)}\right) + r_t^b(i)\left(\frac{cnt_t^b(i)}{cnt_t^a(i) + cnt_t^b(i)}\right) + \Omega_a \tag{10}$$

$$\hat{r}_t^b(j) = (r_t^b(j) - \Omega_b)\left(\frac{cnt_t^b(j)}{cnt_t^a(j) + cnt_t^b(j)}\right) + r_t^a(j)\left(\frac{cnt_t^a(j)}{cnt_t^a(j) + cnt_t^b(j)}\right) + \Omega_b \tag{11}$$

2.2.4 Type IV

The information exchange and update rules for Type IV are the same as Type III, but the updates are only performed for any state i if $abs(r_t^b(i) - r_t^a(i))$ is bounded above by the value d_bound. This threshold ensures that agents with very different estimates are not significantly changing each other's estimates. This addresses our concern that agents with very poor estimates are negatively impacting those that are estimating well. Conversely, however, this will prevent the agents with good estimates from helping those with poor estimates.

2.2.5 Type V

Agents a and b exchange their estimates of all states, r_t^a and r_t^b. The estimates are combined in the same manner as communication type III, but for all states instead of just the current state each agent is in. This is closer to a centralized solution in which state estimates are global information except that inthis information is shared only between pairs of agents each time step.

2.3 Test Environments

In an effort to verify and extend the results achieved in [2] we used the same environment for our experiments and added an additional environment to test penalty-driven learning. The 5x5 Hop World, shown in Figure 1, is an environment in which all agents choose an action at each time step and are given a reward for entering a state. At each time step an agent can choose to move in one of four cardinal directions, but any movement from state 6 or 16 will always move the agent to state 10 and 18, respectively. Most states provide 0 reward but the two states that automatically move the agent provide a reward of 20 and 10, respectively. Each agent's perception is limited to only its own state and is not affected by the location of the other agents in the world.

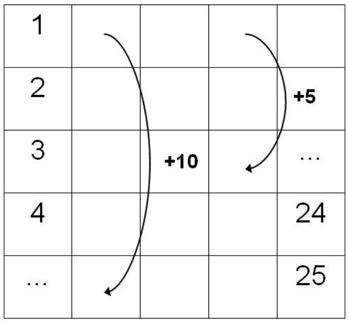

Figure 1: 5x5 Hop World environment

Unlike the Hop World, the Cliff World has only one goal state, but has multiple cliff states that will incur a penalty on the agent. In this environment the optimal path is to walk along the edge of the cliff, but a safer route with less risk is available.

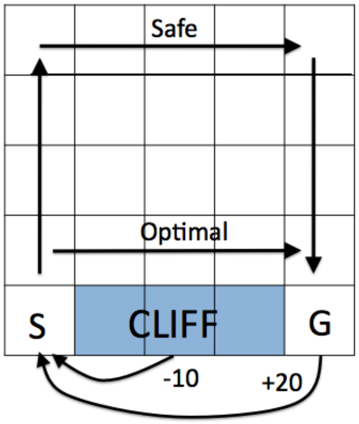

Figure 2: 5 x 5 Cliff World environment

3 Experimental Results

Each communication scheme was executed on both test environments and averaged over 30 runs.

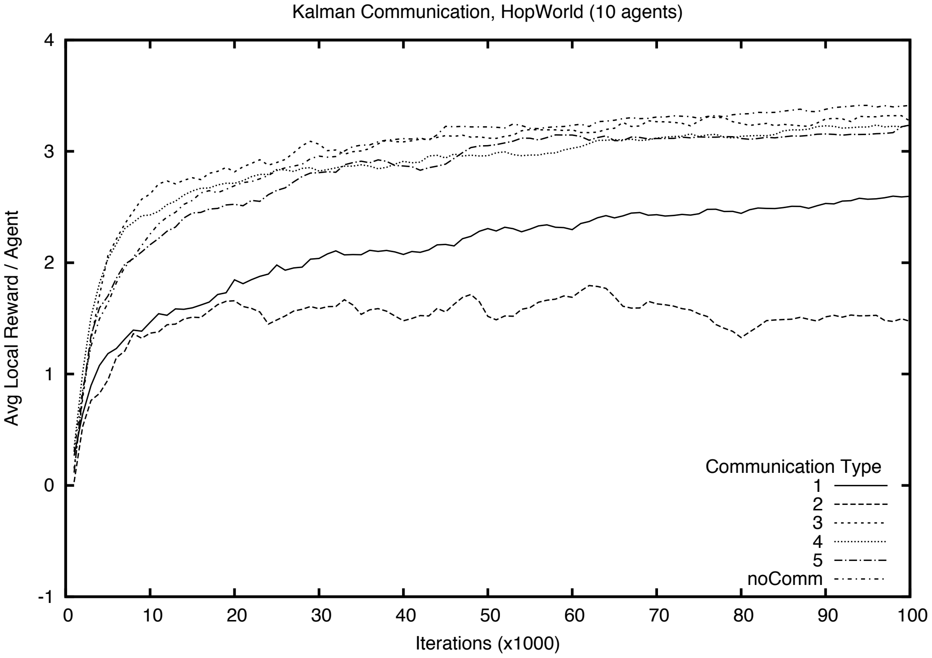

Figure 3: Performance of communication schemes on HopWorld (30 runs)

Communication types I and II performed poorly, doing worse than the fully distributed approach without communication. All other methods performed approximately same, confirming our intuition that the HopWorld environment provides so many options for good performance that the fully distributed approach is able to quickly find a good solution.

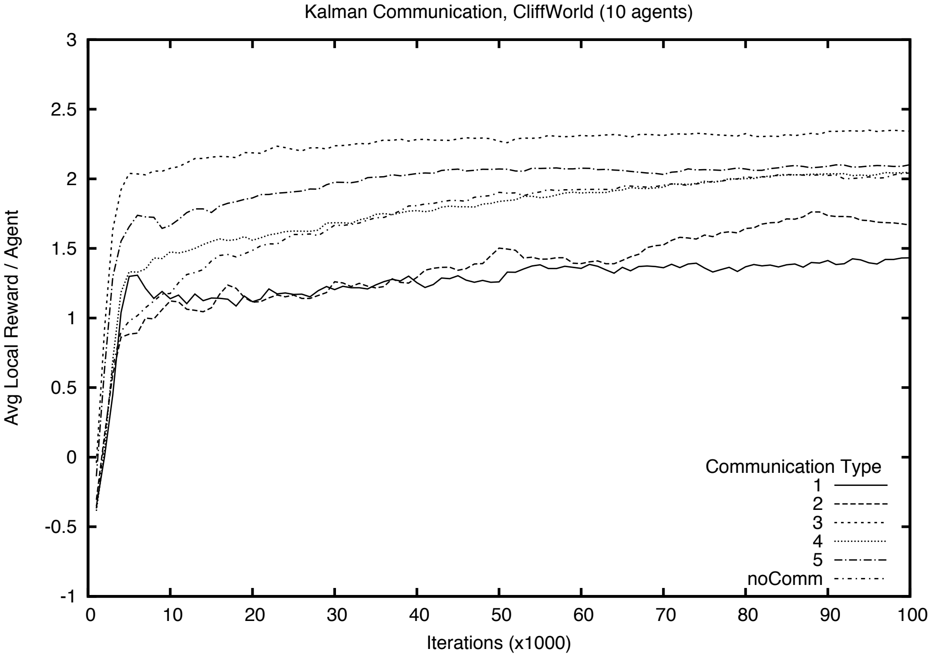

Figure 4: Performance of communication schemes on CliffWorld (30 runs)

Types I and II still perform poorly, but surprisingly Type III, which exchanges information about only the states each agent is in, exceeds the performance of Type V, which exchanges information about all states. Because the estimates are combined based on the number of times each agent has visited a state, we believe that this is caused by the fact that the learners tend to visit good states more often than those it has a poor estimate for. Therefore, in Type III paths that are deemed to be good will get more updates and generate better estimates. On the other hand, Type V exchanges information about all states, including those that have few visits and poor estimates. As a result, states that are not along the good path are more likely to be updated with poor estimates that may overvalue them.

4 State Value Estimation

While the performance of the learners is the focus of this work, it can safely be assumed that the performance of the estimation itself is of greater importance and will directly influence the learning capabilities. In fact, it could be argued that these two components, the state value estimation and the learning algorithm, should be evaluated separately. Therefore, we have started experimenting with a more simple simulation that randomly generates a set of state values and provides the mean squared error of the estimates as the metric for performance. This should help to reduce the noise introduced by the performance of the learning algorithms. Further work on this simulation will allow us to measure effectiveness across a wider range of reward distributions.

Since the Kalman filter proved to be a moderately effective fully distributed solution we also implemented a fully centralized solution so that we could define the upper and lower bounds of performance. The centralized approach uses a simple linear estimation to determine the values of the states and converges to a very small mean squared error for state value estimates within less than 100 time steps. We intend to look at different mechanisms for generating state estimates when some of the states are hidden to the estimator, starting with one state and increasing the number of hidden states gradually. This will provide a different perspective on the state value estimation problem that may actually lead to a similar or the same approach as the Kalman filter.

5 Conclusion

While the Type III communication seems to provide the best performance, the relative success is sensitive to the learning environment and may prove to be ineffective in other types of worlds. The observation that Type III exceeds the performance of Type V in the CliffWorld environment is unexpected and warrants further investigation. It would seem intuitive that sharing information about all states would exceed the performance of sharing only information about the current state each agent is in. We also tried requiring more visits to a given state before allowing an estimate to be shared, but this decreased performance even further.

This analysis of communication schemes to extend the Kalman filter approach, while an interesting exploration of environments that have low learnability, is still not practical. Before any learning can take place the global reward still has to be communicated to each agent, which is infeasible in environments with a large number of agents. However, it should be the case that successful mechanisms in this centralized reward environment should extend to situations where there are rewards based on local or smaller coalition performance.

References

[1] A. K. Agogino. Unifying temporal and structural credit assignment problems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems*, pages 978–985, 2004.

[2] Y. han Chang, T. Ho, and L. P. Kaelbling. All learning is local: Multi-agent learning in global reward games. 2003.

[3] R. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82:35–45, 1960.

[4] S. Sen, I. Sen, M. Sekaran, and J. Hale. Learning to coordinate without sharing information. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pages 426–431, 1994.

[5] R. S. Sutton and A. G. Barto. *Reinforcement Learning: an introduction.* 1998.

[6] K. Tumer and D. Wolpert. A survey of collectives. In *IN COLLECTIVES AND THE DESIGN OF COMPLEX SYSTEMS*, pages 1–42. Springer, 2004.

[7] R. Wright, J. Hudack, N. Gemelli, S. Loscalzo, and T. K. Lue. Agents technology research. Technical report, Air Force Research Laboratory, 2010.

6 Symbols, Abbreviations and Acronyms

COIN - COllective INtelligence
KF - Kalman Filter
MDP - Markov Decision Process
RL - Reinforcement Learning